

Appel de service - Ajout de pièces PWP

Sommaire

Introduction.....	3
CONTEXTE.....	3
OBJECTIF DU DOCUMENT	3
Ajout de pièces.....	3
PÉRIMÈTRE D'APPLICATION.....	3
DESCRIPTION DE L'APPEL DE SERVICE.....	4
Deploiement	7

INTRODUCTION

CONTEXTE

Le GIP SNE développe actuellement un portail web à destination des professionnels partenaires : le PWP.

Ce portail doit permettre de centraliser différents services proposés aux acteurs professionnels de la demande de logement social.

Dans ce cadre, à partir de Mars 2020, le portail sera progressivement mis à disposition des partenaires avec la mise en service du lot numérisation.

De plus les Bordereaux de transmission et Fiches récapitulatives vont fusionner et il y aura un seul document à transmettre : la **Fiche de transmission**, obligatoire pour tout envoi papier.

Les documents peuvent être envoyés soit par voie papier à Docaposte soit par Webservice au PWP, qui les relaie au prestataire de numérisation Docaposte. C'est le WS d'envoi de pièce vers le PWP qui est décrit ci-dessous.

OBJECTIF DU DOCUMENT

Ce document vise à présenter l'appel de service qui permettra d'envoyer des pièces justificatives au PWP pour qu'elles soient transmises au prestataire de numérisation.

AJOUT DE PIÈCES

PÉRIMÈTRE D'APPLICATION

Les principaux utilisateurs sont **les guichets** qui gèrent les demandes de logement social.

Les guichets pourront ajouter des pièces justificatives aux demandes de logement social (DLS) existantes par appel de service.

DESCRIPTION DE L'APPEL DE SERVICE

Ce WS permet de transmettre un pli par WS au PWP. Les pièces justificatives (PJ) seront transmises dans la requête. Chaque appel peut contenir plusieurs PJ mais elles doivent toutes faire partie de la même DLS (même numéro unique). C'est donc de l'ajout **mono-dossier**.

- WS exposé par : PWP
- Méthode : REST
- Condition d'appel : A la demande d'un guichet pour soumettre des PJ.
- La taille maximale par appel est de 20 Mo.
- Les fichiers seront au format **.pdf ou image/***
- URL d'appel au WS :
 - Préproduction : communiquée ultérieurement
 - Production : communiquée ultérieurement.

Données envoyées sous forme de JSON :

Nom du paramètre	Type	Requis	Description
numUnique	String	oui	Numéro unique du dossier sur 18 caractères
idTransaction	String	non	Identifiant de transaction
guichet	Structure Guichet	oui	Voir ci-dessous <i>Structure Guichet</i>
utilisateur	Structure Utilisateur	oui	Voir ci-dessous <i>Structure Utilisateur</i>
fichiers	Liste de structures Fichier	oui	Autant d'éléments dans la liste que de fichier à transmettre. Voir ci-dessous <i>Structure Fichier</i>

Structure Guichet :

Nom du paramètre	Type	Requis	Description
codeGuichet	String	oui	Code guichet sur 5 caractères
codeSIREN	String	oui	Code SIREN

codesTerritoires	String	oui	Codes territoires
------------------	--------	-----	-------------------

Structure Utilisateur :

Nom du paramètre	Type	Requis	Description
mail	String	oui	Adresse mail de l'utilisateur
nom	String	oui	Nom de l'utilisateur
prénom	String	non	Prénom de l'utilisateur

Structure Fichier :

Nom du paramètre	Type	Requis	Description
content	Base64	oui	Contenu du document encodé en base64
hash	String	oui	Hash du document (SHA-1)
nomFichier	String	oui	Nom du document
idTransactionFichier	String	oui	Identifiant de transaction du document

Cet appel de service suivra les caractéristiques suivantes :

- Appel de service de type [POST] /document
- Données dans le POST (body) : application/JSON

Exemple de JSON transmis pour 2 documents :

```
{
  "numUnique": "AAAGGGGG1122166890",
  "idTransaction": "hghuiuhfz21",
  "guichet": {
    "codeGuichet": "ABCDE",
    "codeSIREN": "76889999999997",
    "codesTerritoires": "087"},
  "utilisateur": {
    "mail": "michel.jackson@thefive.com",
    "nom": "Jackson",
    "prenom": "Michel"},
  "fichiers": [{
    "nomFichier": "carteidentite.pdf "
    "content": "JVBERi0xLjQKJeLjz9MKNCAwIG9ixxx....xxx"
```

```

    "hash": "45418a063ac79a2c9cac16b8dfxxx...xxxx684c9e1c13650d11",
    "idTransactionFichier": "a063ac79a2c9cac16b8d"},
  {
    "nomFichier": "carteidentite2.pdf"
    "content": "JVBERG6EdU_K9fH9MKNCAwIG9ixxx...xxx"
    "hash": "3ac79a2c9cac145418a066b8dfxxx...xxxxc13650d11684c9e1",
    "idTransactionFichier": "c9cac145418a066b8d"
  }
}

```

- Retour :

Objet	Type	Description
status	Chaîne	"success" si le requête s'est bien passée ou "error" en cas d'échec
idTransaction	String	Pour identifier à quel envoi correspond un retour, on renverra l'identifiant de transaction si ce champ était présent dans l'envoi.

- Gestion des erreurs

En cas d'erreur, une erreur HTTP sera retournée par l'api :

Code	Description
401	Unauthorized
403	Forbidden
405	Method Not Allowed
415	Unsupported Media Type
422	Unprocessable Entity
501	Not implemented

Ci-après les différents codes d'erreur possibles :

Code	Description
0001	Erreur interne
0002	Echec d'authentification
0003	Paramètres manquants
0004	Paramètres invalides

0009	Erreur lors de l'upload du document
0010	Taille du document incorrect
0011	Hash du document incorrect
0018	Le certificat d'authentification est introuvable
0019	Paramètres du guichet incomplets
0020	Guichet non référencé

DEPLOIEMENT

Chaque éditeur prendra contact avec le GIP SNE (assistance@gip-sne.fr) afin de réaliser des tests d'accrochage avec les service mis à disposition.